

Software, Programming, and Run-Time Coordination for Distributed Robotics

Gruppen, Hanson, Ramamritham
Computer Science Department
University of Massachusetts

Control Structure for Conditioning Learning and Adaptation

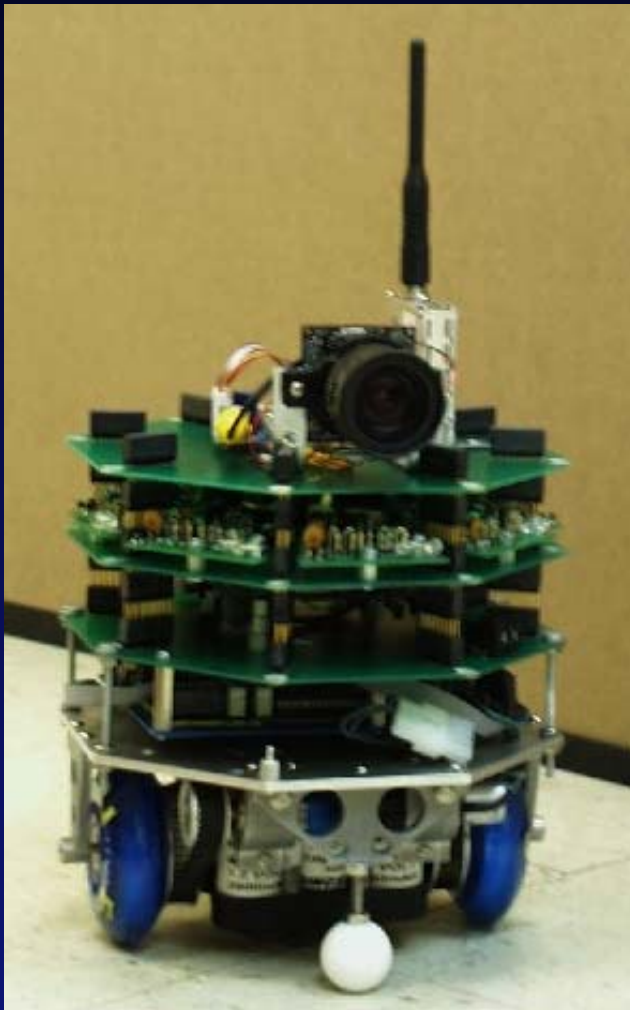
- ✓ exploiting redundancy in distributed, multi-robot systems
null space control interaction
- ✓ interaction-based representations
state estimation from process dynamics
- ✓ *developmental* programming paradigm
constraint-based control composition
physiology - single robot – pairs of robots – teams of robots

Growing a Distributed Controller

Outline:

- UMass UBot – hardware configuration, motor unit
- Individual Behavior
 - motor skills: time, energy, precision path tracking policies
 - concurrent control - “subject-to” null space operator
 - perceptual skills: learning to recognize environmental features
- Coordinated Behavior – exploiting redundancy
 - equivalent 2-way coordination controllers
 - *n*-way coordination configurations

UMass UBot



July 16-18, 2001



- Sensor Layer: vision, (pyrometer, acoustic)
- IR Layer: obstacle detection, inter-robot communication
- Brain board: DIMM PC, RT kernel, I²C, embedded Scenix
- Power Layer: regulation, motor drivers
- Mobility Layer: batteries, motors, drive train

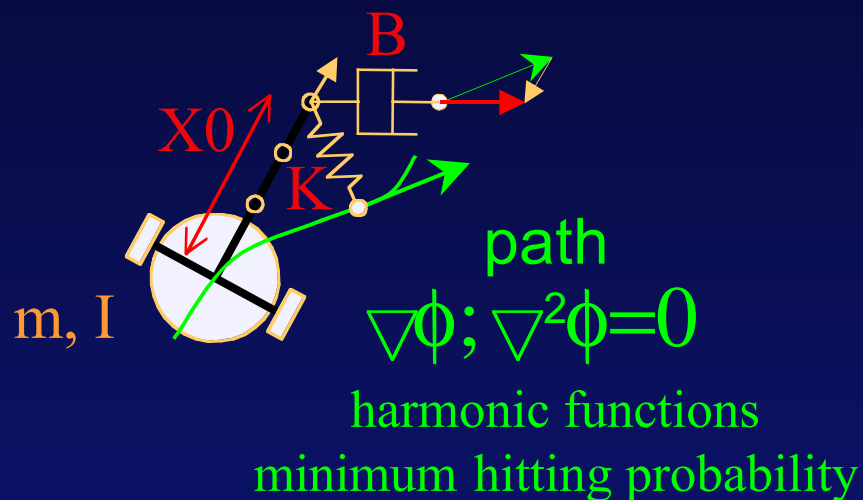
SDR

UBot Control Knowledge

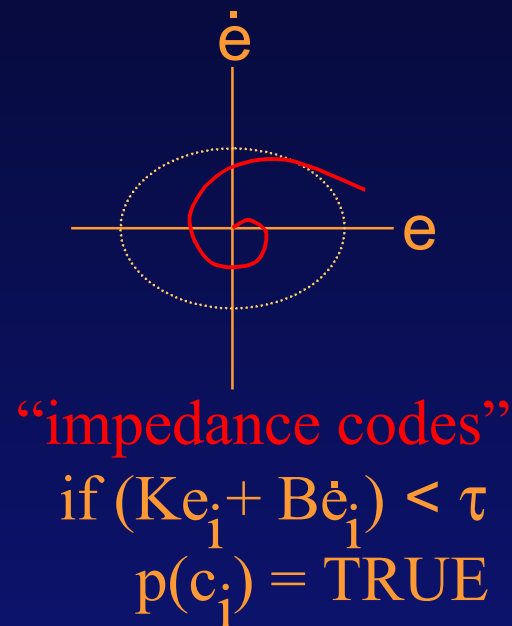
Impedance Control Policies

parametric control

$$c_i(X_0, K, B)$$

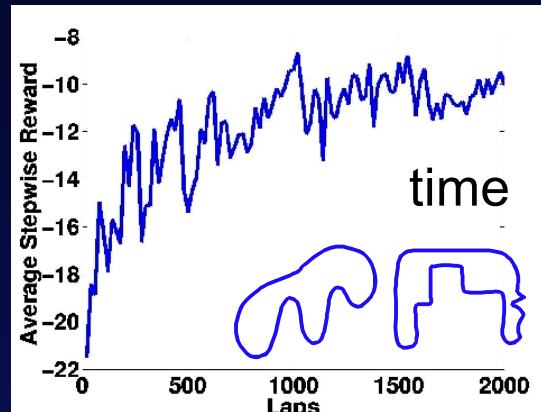
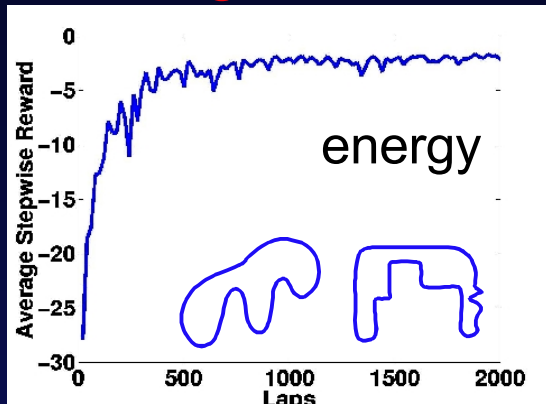


interaction-based
state representation

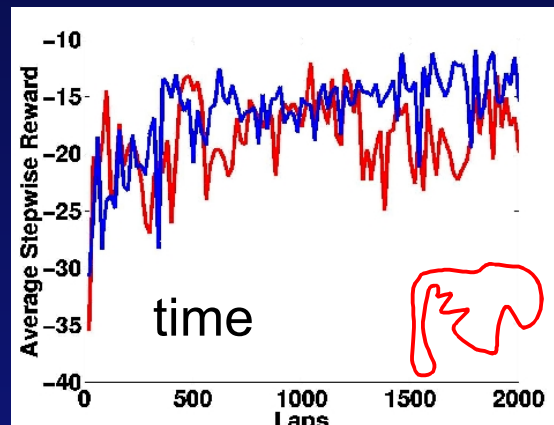
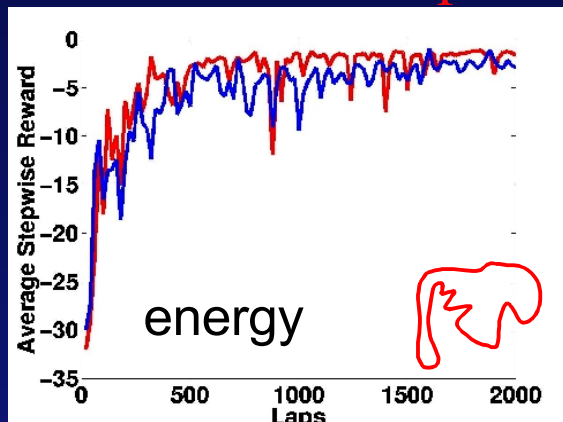


Learning Performance

training set



evaluation: independent test track



ϵ -greedy training ($\epsilon=0.1$)

stepwise reward:

time: $r = -\Delta t / \Delta s$

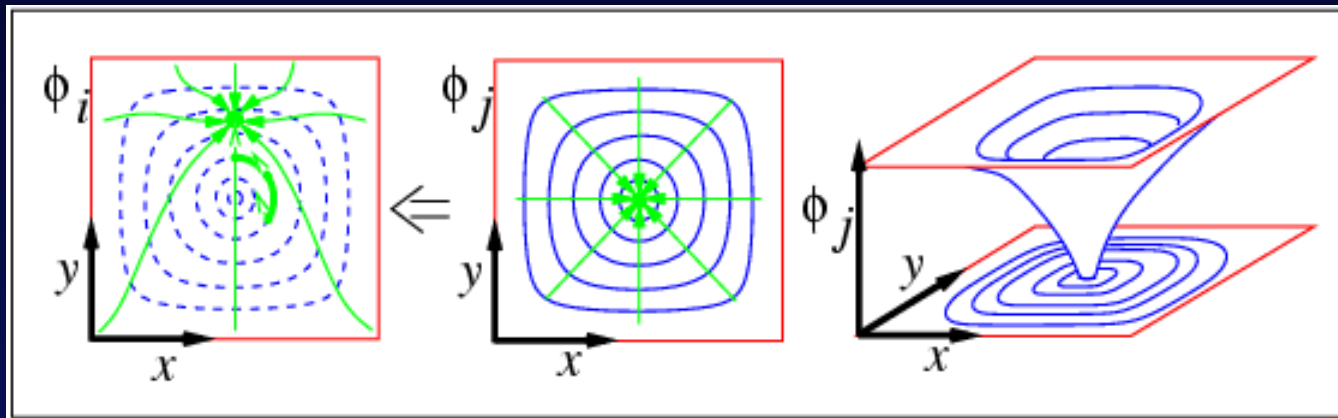
energy: $r = -\Delta E$

greedy evaluation: 20 laps
plotted avg reward: 10 laps

Multi-Objective Controllers

Null Space Control Interactions

$$\phi_i \triangleleft \phi_j$$



$$\Phi_{\text{time}} \triangleleft \Phi_{\text{precision}}$$

$$\Phi_{\text{energy}} \triangleleft \Phi_{\text{precision}}$$

precision - *aka* minimum
hitting probability

Motor Interaction: Parametric Path Control

R denotes a set of robots with proprietary CPUs, effectors, sensors, and communication

Goals: $g(j)$ denotes a goal set derived from robot j (observed or communicated)

ϕ *harmonic potential*: minimum hitting probability

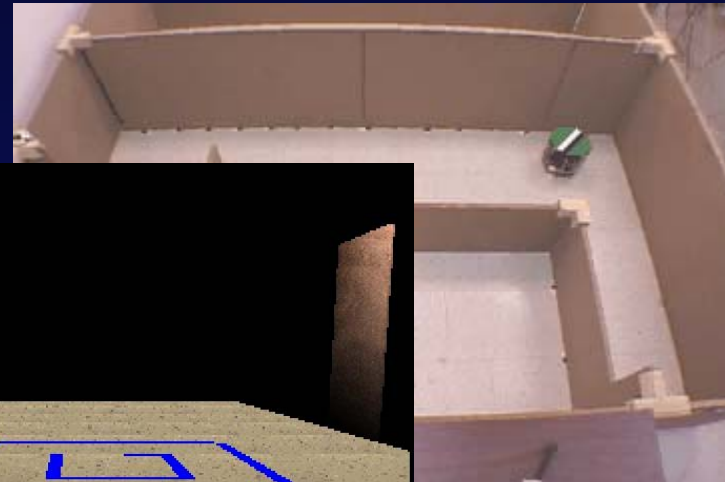
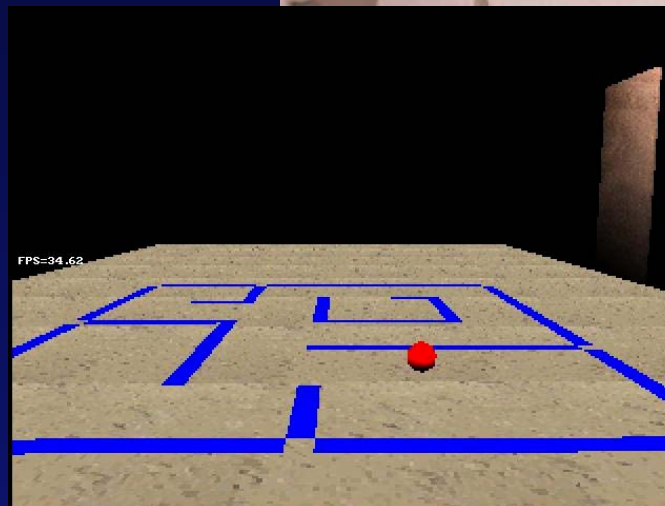
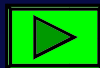
control basis: $\{\phi_i^{g(j)}\} \quad i, j \text{ in } R$

$p(\phi_i^{g(j)})$ evaluates TRUE if robot i is in goal set $g(j)$

Example: Path Control Parameterization

search behavior $\phi_{\text{robot } i}^{\text{world}}$

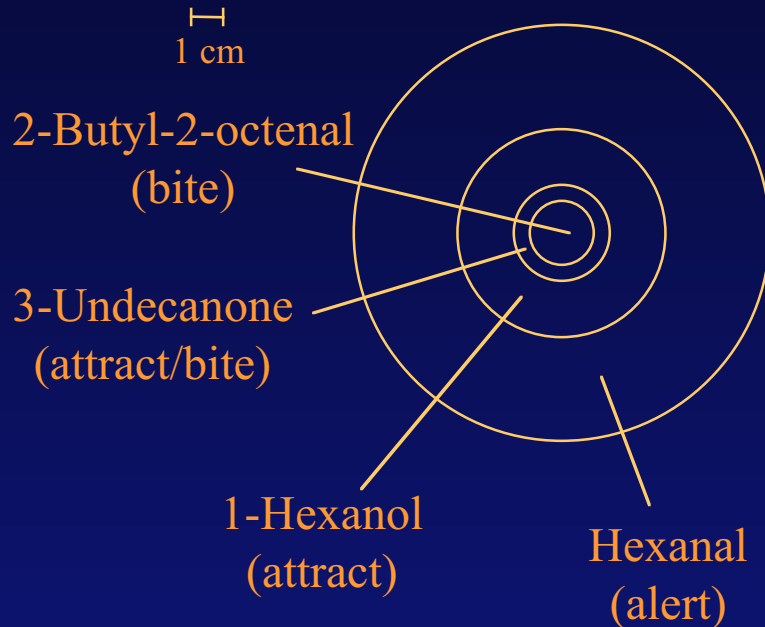
1 robot search simulation



Distributed Behavior

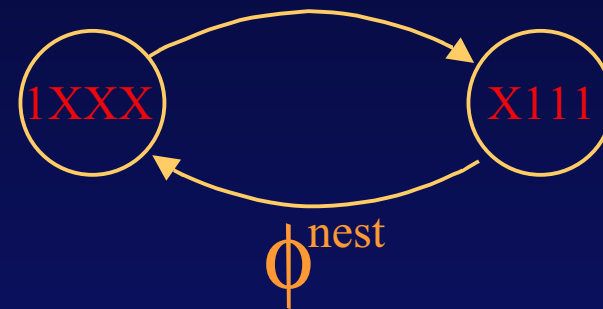
Attack/Repel

Oecophylla longinoda
(African weaver ant)



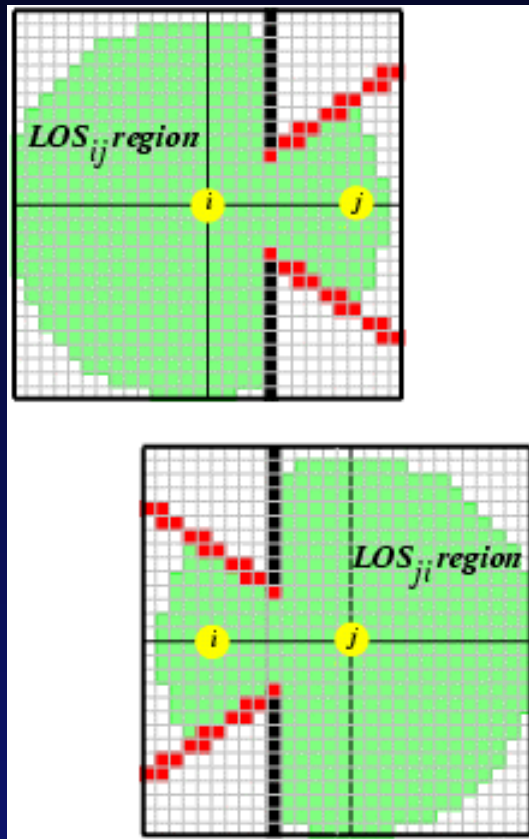
$\mathbf{p} = [\text{nest food alert bite}]$

$\text{RND} \triangleleft \phi^{\text{food}} \triangleleft \phi^{\text{alert}} \triangleleft \phi^{\text{bite}}$



swarm simulation

Push/Pull Kinematic Constraints




if robot j is seeking an external goal:

“push” relation

$$\phi_j^g \triangleleft \phi_j^{g(i)}$$

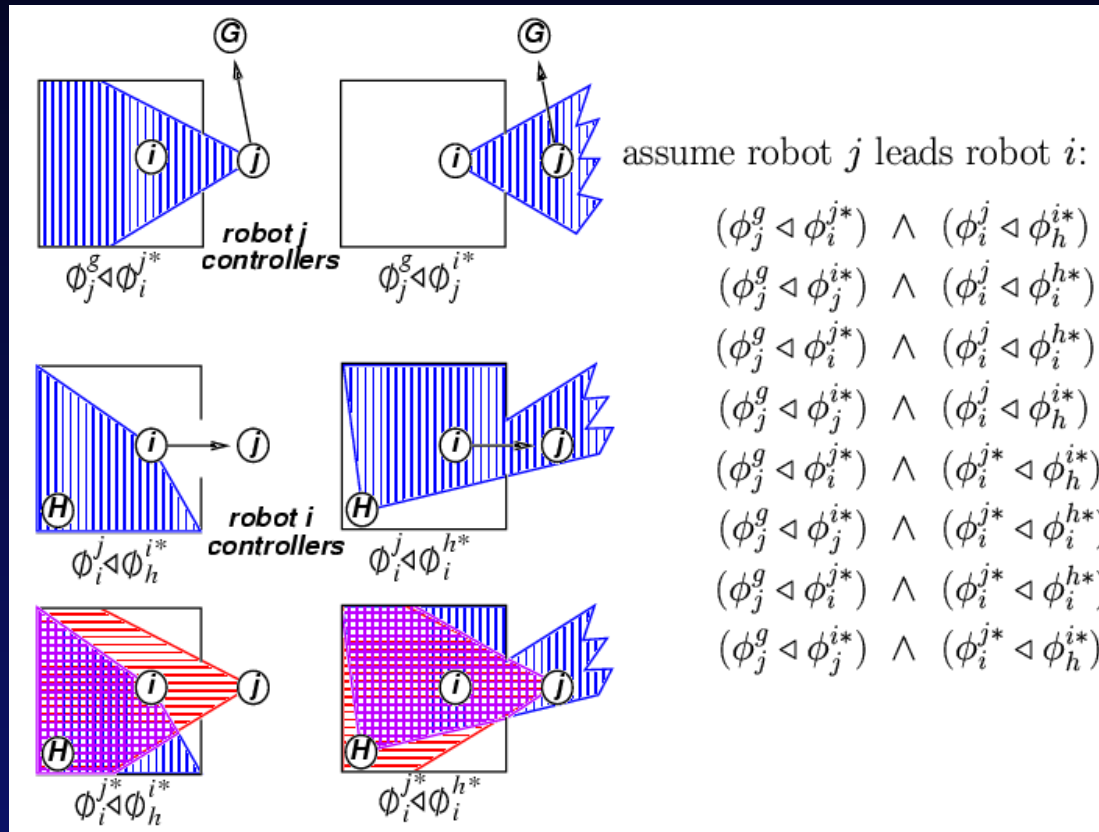
$$\phi_{robot\ i}^{world} \triangleleft \phi_{robot\ j}^{g(i)}$$

2-robot search simulation
(IR-based) 

“pull” relation

$$\phi_j^g \triangleleft \phi_i^{g(j)}$$

Example: Legal 3-Robot Controllers



equivalence class of *correct* controllers with varying quality – CPU, comm bandwidth, power, time

Kinematic Condition

line-of-sight network connectivity

$$G_k^n = \begin{bmatrix} \phi_0^{\text{los}_0} & \phi_0^{\text{los}_1} & \cdot & \cdot & \cdot & \phi_0^{\text{los}_k} \\ \phi_1^{\text{los}_0} & \phi_1^{\text{los}_1} & \cdot & \cdot & \cdot & \phi_1^{\text{los}_k} \\ \cdot & \cdot & & & & \cdot \\ \cdot & \cdot & & & & \cdot \\ \cdot & \cdot & & & & \cdot \\ \phi_k^{\text{los}_0} & \phi_k^{\text{los}_1} & \cdot & \cdot & \cdot & \phi_k^{\text{los}_k} \end{bmatrix}^n$$

$p(G) = \text{TRUE}$ when
all elements $g_{ij}=1$

G defines the equivalence class of “network preserving” control options

Example: Search and Bounded Overwatch

$$\phi_{\text{robot } i}^g \triangleleft G_5^4$$

5-robot search simulation

 (vision-based)

an equivalence class including:

- all permutations of robots,
- all network topologies, and
- all pairwise push/pull configurations

- localization at 7-15 Hz
- precision to 0.15m at 3.5m range
- baseline variation 1-2m

$$G_3^2$$



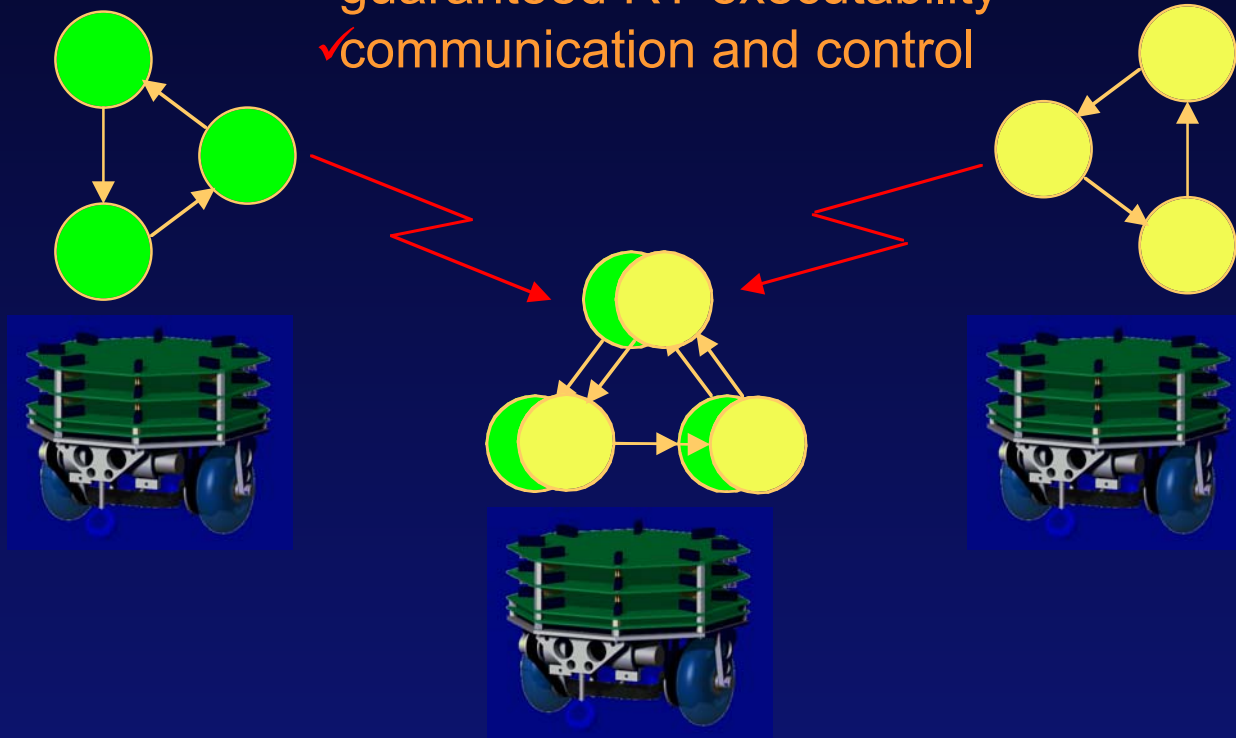
bounded overwatch

Stable Real-Time Adaptation

.be extensions for real-time control

✓guaranteed RT executability

✓communication and control



Deliverable UMass Technology

UBot implementation

- hardware/firmware implementation
- .be code – RT control/adaptation, graphical programming
- concurrent, multi-initiative policies

Distributed, Multi-Robot Control

- generalized null space – “subject-to” compositional framework
- structure for learning to optimize multi-robot behavior
- policies for search and mapping, leader-follower, network connectivity, bounded overwatch
- reconfigurable “virtual” sensors and coordinated mobility